A rest stop on the unending road to provable security

Provable security, where system security properties have been mathematically proven, requires generally accepted assumptions. However, these assumptions are often unproven, which makes security conditional: it is guaranteed only when the unproven assumptions happen to hold. Professor Virgil Gligor, of Carnegie Mellon University, demonstrates the importance of an unconditional solution to any security or cryptography problem. He also shows how software root of trust establishment can be carried out with a simple verifier unconditionally, with no secrets, trusted hardware modules, or adversary bounds, providing a practical rest-stop to provable security.

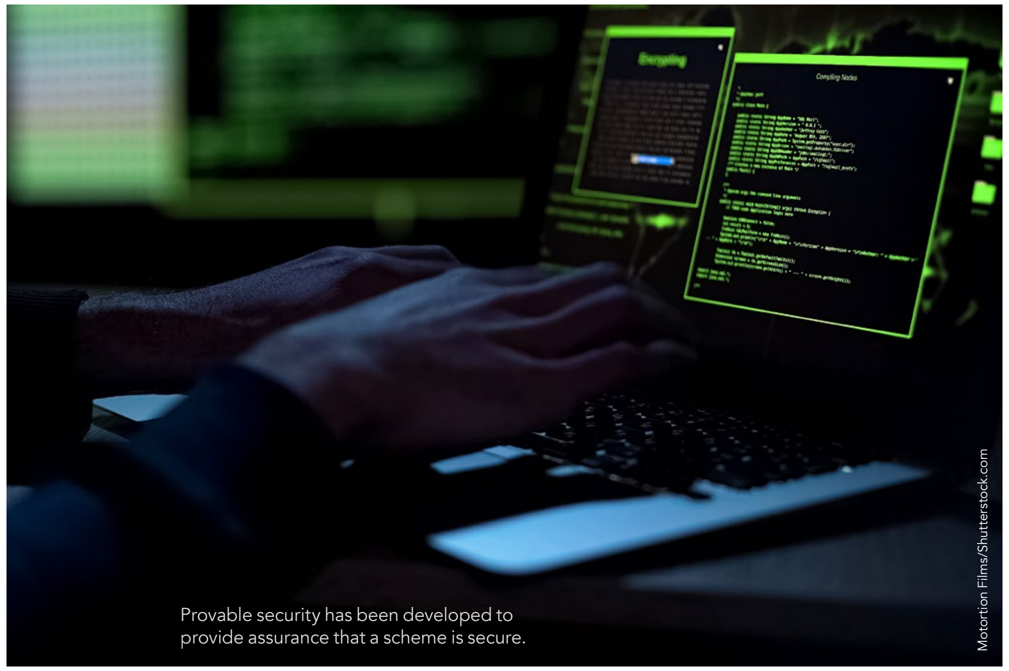
No longer restricted to ordinary computers, software is embedded in machines and devices that impact every aspect of our lives. In contrast to physical systems, software is continually modified via updates and patches that are delivered through software supply chains. This exposes it to a multitude of flaws, both unintentional and malicious. Cyber-attacks, targeting software developers and suppliers, with the aim of accessing source codes, build processes and update mechanisms, are on the rise. Moreover, supply chain attacks have emerged with hackers infecting legitimate apps in order to distribute malware. Consequently, the demand for computer security that can protect software from these threats is escalating; but how is security guaranteed?

ErikStudio/Shutterstock.com

WHAT IS SECURE?

It has been shown that our perception of what is secure and what is not is poor. A cryptosystem is a suite of cryptographic algorithms required to implement a particular security service to achieve confidentiality and authenticity of data by encryption. Cryptography history is littered with examples of cryptosystems that were initially thought to be highly secure but were later found to fail. Essentially, there is no way of testing if a cryptographic algorithm is secure. Instead, provable security has been developed to provide assurance that a scheme is secure. Provable security means that the security properties of a system have been mathematically proven under generally accepted assumptions. Security cannot be proven, however, without making often unproven assumptions about other system properties, making system security conditional. For example, can the secrecy of a cryptographic key be maintained without requiring trusted hardware modules assumed to be securely maintained and used or limiting the adversary's computing power? Can software execution be guaranteed on computers whose device firmware is compromised without relying on trusted hardware modules? Can a cryptographic wallet demonstrably maintain content secrecy and authenticity without limiting adversary's computing power? A negative answer to any of these questions shows why security is conditional in practice and hence more costly to maintain and use.

Professor Virgil Gligor, of Carnegie Mellon University, explains why an unconditional solution to any security or cryptography problem is important. He also contends that software root of trust can be established with a simple external verifier. A root of trust is the foundational security element within a cryptographic system that can always be trusted. In a breakthrough study in 2019, he demonstrated how a software root of trust in an untrusted system can be ascertained unconditionally, with no secrets, trusted hardware modules, or adversary bounds. He also explains how the verifier's trustworthiness can be proven without any dependencies on other unverified computations.



Provable security has been developed to provide assurance that a scheme is secure.

Motortion Films/Shutterstock.com

MALWARE IMPLANTS

Hackers can furtively implant malicious software, or malware, into a system's firmware. System firmware contains the most privileged program codes that control the system's hardware operation. Compromises within a supply chain enable malware implants to corrupt firmware. This can occur before or after a system is delivered to an unsuspecting user. Malware implants can also take place when an adversary compromises the updates that firmware applies to itself via out-of-band protocols that use alternate access methods into a network. It is also possible for an adversary to acquire physical access to a system for a few minutes, providing an opportunity for them to insert a malware-loaded USB device into a system. It is anticipated that by 2022, 70% of organisations without firmware upgrade plans will suffer breaches due to firmware vulnerability.

UNDETECTABLE

Without taking a system apart, malware operation in firmware can often remain undetected. Malware can survive in device controllers (such as network interface cards, disk, and baseboard-management controllers) in spite of repeated power cycles, secure re-boots, and operating system re-imaging. They can even remain unnoticed when commercial anti-malware tools and system monitors are employed. Malware can communicate with remote controllers via covert channels, or by ostensibly legitimate measures, and exploit operating system vulnerabilities. It can hide in areas that do not get updated. Malware can also respond to naive attempts of re-flashing firmware to remove them with fake notifications, such as 'update complete' or 'already the latest version'.

DETECTION

If persistent malware remains undetected with no tell-tale signs of its presence, how can an external observer determine whether there is or isn't malware in a system's firmware, without taking the system apart? To detect malware, Professor Gligor proposes that a trustworthy external verifier is employed to challenge the system with the execution of special functions. Then this verifier measures the system's responses and determines if they are correct and timely. If they are, the system is considered to be malware free. If not, the verifier detects that malware has been executed or some unaccounted malware content is present.

ESTABLISHING ROOT OF TRUST UNCONDITIONALLY

Root of trust establishment is required for a number of basic system security issues, such as starting a system in a secure initial state and performing trusted recovery. Establishing root of trust assures the system has all of the content chosen by a trusted verifier, and only the chosen content, nothing

> **Given sufficient computation power most cryptographic protocols can be broken, including secret-less ones.**

It is anticipated that by 2022, 70% of organisations without firmware upgrade plans will suffer breaches due to firmware vulnerability.

Cherries/Shutterstock.com

Rawpixel.com/Shutterstock.com

In his research, Professor Gligor demonstrates how to establish root of trust unconditionally, ie, without trade-offs.

malware freedom unconditionally is important because the disclosure of secrets can be bought, coerced, or acquired by brute-force, even when they are hidden within hardware. Professor Gligor reminds us that "given sufficient computational power most cryptographic protocols can be broken, including secret-less ones".

A PRACTICAL REST-STOP TO PROVABLE SECURITY

Professor Gligor demonstrates extra. Obtaining such a guarantee is challenging because malware can survive repeated secure- and trusted-boot operations as well as avoiding detection by anti-malware tools. In his research, Professor Gligor demonstrates how to establish root of trust unconditionally, ie, without trade-offs.

CHALLENGE FUNCTION

In his recent publications, Professor Gligor establishes that an external verifier can provably establish persistent-malware freedom when the challenge functions, ie, the mathematical functions that prove security, are a particular type of polynomials. These special polynomials are *k*-independent randomised polynomials, and he has argued that it is rather unlikely that more effective challenge functions exist.

These polynomials are used to construct *k*-independent universal hash functions. Hash functions are used to access data in data storage and retrieval applications and are efficient both in terms of computation and storage space. Universal hashing involves the selection of a hash function at random from a family of hash functions with a particular mathematical property. A family of hash functions is *k*-independent if it can be guaranteed that when a function is selected at random from that family, the hash codes of any designated number *(k)* of keys are independent random variables.

Professor Gligor shows that, because these polynomials have a unique optimal execution time in a particular memory space, there is no other program that can execute them in less time and space on a particular system. Furthermore, no other function, or input, can returns the same result that these polynomials do. Root of trust establishment is therefore possible with a simple external verifier.

solarseven/Shutterstock.com

MALWARE FREEDOM

The malware-freedom test based on *k*-independent randomised polynomials is unconditional in a most general sense. It does not require secrets. There is no requirement, therefore, for hardware security modules or tokens to protect secrets or cryptographic codes. Moreover, it does not assume any limit on the adversary's power. Establishing the necessity of external verifiers to achieve unconditional malware freedom. They proffer a provable advantage to a defender over any adversary, and they can outlive technology advances, making them valuable for post quantum computing. The malware-freedom test offers the first sufficient and unconditional solution to the problem of root of trust establishment. To date, no other security or cryptography problem has been unconditionally solved in this general sense. Professor Gligor observes that this unconditional malware freedom provides a "rest-stop on the unending road to provable security against any adversary".

Unconditional malware freedom provides a "rest-stop on the unending road to provable security" against any adversary.

Behind the Research

Virgil Gligor

E: gligor@cmu.edu **T:** +1 412-268-9833 **W:** https://www.ece.cmu.edu/directory/bios/gligor-virgil.html
W: https://www.cylab.cmu.edu/directory/bios/gligor-virgil.html **W:** wikipedia.org/wiki/Virgil_D._Gligor

Research Objectives

Virgil Gligor's research addresses problems of trustworthy computing in the presence of an active adversary, for example, malware and malicious insiders.

Detail

Address
Carnegie Mellon University, 4720 Forbes Avenue, Pittsburgh PA 15213 USA

Bio
Virgil Gligor is a Professor at Carnegie Mellon University. He received the 2006 *National Information Systems Security Award* given by NIST and NSA, 2011 *Outstanding Innovation Award* of the ACM SIGSAC, and 2013 IEEE Computer Society *Technical Achievement Award*. He was inducted into the Cybersecurity Hall of Fame in 2019.

Funding
Carnegie Mellon University

References

Gligor, VD (2020). A Rest Stop on the Unending Road to Provable Security (Transcript of Discussion). In: Anderson J., Stajano F., Christianson B., Matyáš V. (eds) Security Protocols XXVII. *Security Protocols 2019*. Lecture Notes in Computer Science, vol 12287. Springer, Cham. https://doi.org/10.1007/978-3-030-57043-9_22

Gligor, V & Woo, M (2019). Establishing Software Root of Trust Unconditionally, In *Proceedings of the Network and Distributed Systems Symposium* (NDSS), San Diego, CA. Feb. 2019 (Full paper: CMU - CyLab - Technical Report 18-003, Nov., 2018). https://www.ndss-symposium.org/wp-content/uploads/2019/02/ndss2019_10-4_Gligor_paper.pdf

Gligor, V (2019). Winning Against any Adversary on Commodity Computer Systems. In *Proceedings of the 1st ACM Workshop on Workshop on Cyber-Security Arms Race* (CYSARM'19). Association for Computing Machinery, New York, NY, USA, 1–2. https://doi.org/10.1145/3338511.3357346

Personal Response

Why do you think that no other security or cryptography problem has been unconditionally solved in this general sense to date?

" Assuming away unconditional-security requirements is very tempting as it can greatly reduce system development cost and time. Unfortunately, it allows adversary attacks against the assumptions made, and security becomes more expensive to use and maintain. Assuming secrecy of encryption keys that are selected uniformly at random and never reused implies that an adversary's computation power can be unbounded. This often requires trusted hardware for key-secrecy protection and higher security management cost, skills, and aligned interests. Assuming that the adversary's power can be bounded enables public-key cryptography at the higher cost of having to maintain the security of fragile public-key infrastructures. "

CyLab Carnegie Mellon University
Security and Privacy Institute

Peera_stockfoto/Shutterstock.com